



# Watersheds on weighted graphs

Fernand Meyer

## ► To cite this version:

Fernand Meyer. Watersheds on weighted graphs. Pattern Recognition Letters, 2014, 47, pp.72-79. 10.1016/j.patrec.2014.02.018 . hal-01111752

**HAL Id: hal-01111752**

**<https://hal-mines-paristech.archives-ouvertes.fr/hal-01111752>**

Submitted on 30 Jan 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution| 4.0 International License

# Watersheds for weighted graphs

Fernand Meyer  
Centre for Mathematical Morphology

## Abstract

The watershed is an efficient and versatile segmentation tool, as it partitions the images into disjoint catchment basins. We study the watershed on node or edge weighted graphs. We do not aim at constructing a partition of the nodes but consider the the catchment zones, i.e. the attraction zone of a drop of water. Often, such zones largely overlap. In a first part we show how to derive from a node or edge weighted graph a flooding graph with the same trajectories of a drop of water, whether one considers its node weights alone or its edge weights alone. In a second part we show how to reduce the number of possible trajectories of a drop of water in order to generate watershed partitions.

## 1 Introduction

Catchment basins and watershed lines are topographical notions. A catchment basin is the attraction domain of a regional minimum: a drop of water falling everywhere within this domain glides towards this minimum. The catchment basins are therefore disjoint, and the lines which separate them are called watersheds. To produce such partitions, each point of the space has to be allocated to one and only one basin, excepted along the watershed lines.

The transposition of these concepts to image processing proved to be very efficient for segmentation purposes [11] : the topographic surface being the gradient image of the image to segment. The inside of the objects being expressed by low gradient values in form of regional minima ; the boundaries of the object being expressed by high gradient values. The catchment basins of the gradient image form a partition of the image, where each region contains one and only one regional minimum.

On real topographical surfaces however, the watershed lines may contain thick zones, due to the presence of plateaus and button holes, i.e. large areas from where a drop of water may reach several distinct regional minima. The representation of images on digital grids introduces additional difficulties, due to the discretisation and the anisotropy of a grid. In order to overcome these difficulties, a large number of studies have been published on the watershed, refining the definitions and proposing sophisticated algorithms for producing partitions ([9],[12],[1] ; see [20] for a recent bibliography). The watershed has been studied and used in two contexts:

- \* For images, modelled as node weighted graphs. Reference [21] presents algorithms for watershed computing, and ref. [18] describes how the concept and the algorithms of watershed have been influenced by both theoretical considerations and hardware capabilities.

- \* For edge weighted graphs, in which the nodes represent regions or pixels and the weighted edges represent the dissimilarity between adjacent nodes. The waterfall hierarchy is based on a series of watersheds on such graphs ([2]). J.Cousty et al studied these graphs and called watershed cut the graph obtained by cutting all edges linking two distinct basins ([7]).

Two types of algorithms coexist. Shortest paths algorithm construct the zone of influence of the regional minima for various distance function associated to the topography ([23, 8, 10, 13, 19]). Another class of algorithms are based on thinnings : the topographic surface is progressively lowered in such a way that the regional minima are expanded without merging with each other (S.Beucher et al [3] for geodesic binary thinnings, G.Bertrand [7] for grey tone thinnings based on W destructible points).

All these approaches simultaneously create catchment basins by extending the minima, while creating disjoint basins. We propose below another point of view, radically different, and which opens new horizons. We keep the framework of the finite weighted graphs and take as basis of our study, the flooding paths, i.e. the possible trajectories of a drop of water. A drop of water may reach several regional minima and belong to several overlapping catchment zones (a catchment zone contains the classical catchment basin). The development starts from the flooding adjunction, and reaches the key notion of a flooding graph, which unifies the flooding paths defined by nodes or by edges.

Ordering the flooding paths according to their steepness and suppressing all those with a low steepness reduces (without modifying the node or edge weights) shrinks the catchment zones and reduce the domains where they overlap. A parallel algorithm implements this pruning

of the graph. A final section shows how to extract the downstream or upstream of a node, and by combining them, the catchment zone of this node. Interestingly, no choice intervenes for establishing the catchment zones (1st step), and practically none for generating the watersheds (2nd step).

An early version of this work may be found in [17] and a method for implementing graph algorithms on images in [16].

## 2 The topography of node or edge weighted graphs

A *non oriented graph*  $G = [N, E]$  contains a set  $N$  of vertices or nodes and a set  $E$  of edges ; an edge being a pair of vertices. The nodes are designated with small letters:  $p, q, r...$ . The edge linking the nodes  $p$  and  $q$  is designated by  $e_{pq}$ .

A *path*,  $\pi$ , is a sequence of vertices and edges, interweaved in the following way:  $\pi$  starts with a vertex, say  $p$ , followed by an edge  $e_{pq}$ , incident to  $p$ , followed by the other endpoint  $q$  of  $e_{pq}$ , and so on.

Edges and/or nodes may be weighted. Denote by  $\mathcal{F}_e$  and  $\mathcal{F}_n$  the sets of non negative weight functions on the edges and on the nodes respectively. The function  $\eta \in \mathcal{F}_e$  takes the value  $\eta_{pq}$  on the edge  $e_{pq}$  ; the function  $\nu \in \mathcal{F}_n$  takes the weight  $\nu_p$  on the node  $p$ .

For the same graph  $G$ , we designate  $G(\text{nil}, \nu)$ ,  $G(\eta, \text{nil})$  and  $G(\eta, \nu)$  respectively the node weighted, edge weighted, edge and node weighted graph.

### 2.1 Flooding edges, flooding paths, regional minima and catchment zones

A drop of water falling on a topographics surface follows a flooding path. In order to be sure that each node is linked by a flooding path with a regional minimum, we consider all flooding paths which are physically possible: a drop of water may follow any motion, except flowing upstream.

#### 2.1.1 Node weighted graphs

We consider a node weighted graph  $G(\text{nil}, \nu)$ . A drop of water may quit a node  $p$ , follow the edge  $(p, q)$  and reach the node  $q$  if the nodes  $p$  and  $q$  verify  $n_p \geq n_q$ .

**Definition 1** *The edge  $(p, q)$  is a flooding edge of its extremity  $p$  iff  $n_p \geq n_q$ .*

**Definition 2** *A path  $p - e_{pq} - q - e_{qs} - s...$  is a flooding path if each node except the last one is followed by one of its flooding edges.*

It follows that the weights along a flooding path are never increasing.

**Definition 3** *A regional minimum of  $G_n$  is a maximal connected subgraph, in which all inside nodes have the same weight and all neighboring outside nodes have a higher altitude.*

**Criterion 4** *A node  $p$  belongs to a regional minimum if no flooding path with origin  $p$  leads to a lower node.*

### 2.1.2 Edge weighted graphs

A drop of water at a node  $p$  compares the weights of all edges allowing to quit  $p$  and follows one of the adjacent the edges with the lowest weight.

**Definition 5** *An edge  $(p, q)$  is a flooding edge of the node  $p$ , if it is one of the lowest adjacent edges of  $p$ .*

**Definition 6** *A path  $p - e_{pq} - q - e_{qs} - s \dots$  is a flooding path if each node except the last one is followed by one of its flooding edges*

It follows that the the weights of the successive edges in a flooding path are never increasing (if  $p, q$  and  $r$  are three successive nodes of a flooding path, then  $e_{qr}$  is one of the lowest edges of  $q$  and thus  $e_{pq} \geq e_{qr}$ ).

**Definition 7** *A regional minimum is a maximal connected subgraph of  $G$ , in which all inside edges have the same weight and all edges linking an inside node with an outside node have higher weights.*

The regional minima contain at least two nodes and isolated regional minima do not exist.

**Criterion 8** *A node  $p$  belongs to a regional minimum if no flooding path with origin  $p$  contains an edge which is lower than the edges inside the minimum.*

### 2.1.3 Common properties of node and edge weighted graphs

The definition of flooding edges, flooding paths and regional minima is not the same for node and edge weighted graphs. But given the definition of flooding edges, paths and regional minima, they share in common the following properties. A drop of water only follows flooding paths.

**Definition 9** *The catchment zone of a regional minimum is the set of nodes linked with this minimum through a flooding path.*

Each node of a node or edge weighted graph either belongs to a regional minimum or is linked with a regional minimum through a flooding path.

**Corollary 10** *Each node of a node or edge weighted graph belongs to at least one catchment zone.*

### 2.1.4 Completing the missing weights

**For a node weighted graph** Consider the node weighted graph  $G(nil, \nu)$ . A drop of water may flow through a flooding edge  $e_{pq}$  from  $p$  to  $q$  if  $\nu_p \geq \nu_q$  and in both direction if  $\nu_p = \nu_q$ . Each edge  $e_{pq}$  is a flooding edge as  $\nu_p \geq \nu_q$  or  $\nu_q \geq \nu_p$ . In any case, the weight  $\nu_p \vee \nu_q$  represents the weight of the node from where a drop of water crossing the edge  $e_{pq}$  may come from.

We define the operator  $\delta_{en}$  from  $\mathcal{F}_n$  into  $\mathcal{F}_e : \nu \rightarrow \delta_{en}\nu$ . The function  $\delta_{en}\nu$  of  $\mathcal{F}_e$  takes the value  $(\delta_{en}\nu)_{pq} = \nu_p \vee \nu_q$  on the edge  $e_{pq}$ . Each edge  $e_{ps}$  with extremity  $p$  gets a weight  $\nu_p \vee \nu_s \geq \nu_p$ . Only the flooding edges of  $G(nil, \nu)$ , such as  $e_{pq}$ , verifying  $\nu_p \geq \nu_q$  get the weight  $\nu_p$ . Thus these edges are the lowest adjacent edges of  $p$ , i.e. flooding edges in the graph  $G(\delta_{en}\nu, nil)$ .

**Lemma 11** *Each flooding edge in the graph  $G(nil, \nu)$  also is a flooding edge in the graph  $G(\delta_{en}\nu, nil)$*

The reverse is not true and the graphs  $G(nil, \nu)$  and  $G(\delta_{en}\nu, nil)$  do not have the same regional minima as shown by the counter-example in fig.1.

**For an edge weighted graph** Consider the edge weighted graph  $G(\eta, nil)$ . A drop of water leaving a node  $p$  always follows a flooding edge of  $p$ . We define the operator  $\varepsilon_{ne}$  from  $\mathcal{F}_e$  into  $\mathcal{F}_n : \eta \rightarrow \varepsilon_{ne}\eta$ . The function  $\varepsilon_{ne}\eta$  of  $\mathcal{F}_n$  takes on the node  $p$  the weight of the flooding edges of  $p$ , i.e.  $(\varepsilon_{ne}\eta)_p = \bigwedge_{(r \text{ neighbors of } p)} \eta_{ps}$ .

Consider now a flooding edge  $e_{pq}$  of extremity  $p : (\varepsilon_{ne}\eta)_p = \bigwedge_{(r \text{ neighbors of } p)} \eta_{ps} \geq (\varepsilon_{ne}\eta)_q$  showing that  $e_{pq}$  also is a flooding edge of  $p$  in the graph  $G(nil, \varepsilon_{ne}\eta)$ .

**Lemma 12** *Each flooding edge in the graph  $G(\eta, nil)$  also is a flooding edge in the graph  $G(nil, \varepsilon_{ne}\eta)$*

The reverse is not true and the graphs  $G(\eta, nil)$  and  $G(nil, \varepsilon_{ne}\eta)$  do not have the same regional minima as shown by the counter-example in fig.2.

## 3 The flooding adjunction

The pair of operators  $(\delta_{en}, \varepsilon_{ne})$  form an adjunction :  $\forall \eta \in \mathcal{F}_e, \forall \nu \in \mathcal{F}_n : \delta_{en}\nu < \eta \Leftrightarrow \nu < \varepsilon_{ne}\eta$ . The following properties are classical ([22]).

- $\delta_{en}$  is a dilation from  $\mathcal{F}_n$  into  $\mathcal{F}_e$

- $\varepsilon_{ne}$  is an erosion from  $\mathcal{F}_e$  into  $\mathcal{F}_n$
- $\gamma_e = \delta_{en}\varepsilon_{ne}$  is an opening from  $\mathcal{F}_e$  into  $\mathcal{F}_e$
- $\varphi_n = \varepsilon_{ne}\delta_{en}$  is a closing from  $\mathcal{F}_n$  into  $\mathcal{F}_n$

Furthermore as  $\varepsilon_{ne} = (\varepsilon_{ne}\delta_{en})\varepsilon_{ne} = \varphi_n\varepsilon_{ne}$ , for  $\eta$  in  $\mathcal{F}_e$  the function  $\varepsilon_{ne}\eta$  in  $\mathcal{F}_n$  is closed by the closing  $\varphi_n$ . Symetrically as  $\delta_{en} = (\delta_{en}\varepsilon_{ne})\delta_{en}$  for  $\nu$  in  $\mathcal{F}_n$ , the function  $\delta_{en}\nu$  in  $\mathcal{F}_e$  is opened by the opening  $\gamma_e$ .

These operators belong to a larger family of operators between nodes, edges, faces of planar graphs (in [14] for constructing micro-viscous levelings and filters, in [6] for binary filters). The flooding adjunction has already been used for studying the watershed and waterfall algorithms in ([15]).

### 3.1 Flooding graphs

#### 3.1.1 Identity of the flooding edges in both node and edge weighted graphs ?

Lemma 9 states: each flooding edge in the graph  $G(nil, \nu)$  also is a flooding edge in the graph  $G(\delta_{en}\nu, nil)$ . Replacing  $\eta$  by the function  $\delta_{en}\nu$  in the lemma 10 shows that each flooding edge in the graph  $G(\delta_{en}\nu, nil)$  also is a flooding edge in the graph  $G(nil, \varepsilon_{ne}\delta_{en}\nu)$ . It follows:

**Lemma 13** *In the case where  $\varepsilon_{ne}\delta_{en}\nu = \varphi_n\nu = \nu$ , both graphs  $G(nil, \nu)$  and  $G(\delta_{en}\nu, nil)$  have the same flooding edges.*

Similarly lemma 10 states: each flooding edge in the graph  $G(\eta, nil)$  also is a flooding edge in the graph  $G(nil, \varepsilon_{ne}\eta)$ . Replacing  $\nu$  by the function  $\varepsilon_{ne}\eta$  in the lemma 9 shows that each flooding edge in the graph  $G(nil, \varepsilon_{ne}\eta)$  also is a flooding edge in the graph  $G(\delta_{en}\varepsilon_{ne}\eta, nil)$ . It follows:

**Lemma 14** *In the case where  $\delta_{en}\varepsilon_{ne}\eta = \gamma_e\eta = \eta$ , both graphs  $G(\eta, nil)$  and  $G(nil, \varepsilon_{ne}\eta)$  have the same flooding edges.*

In general however, the node weights  $\nu$  of a graph  $G(nil, \nu)$  are not invariant by  $\varphi_n$  and the edge weights  $\eta$  of a graph  $G(\eta, nil)$  not invariant by  $\gamma_e$ . The next section shows how to edit the graph  $G$  without modifying the weights of the nodes (resp. edges) so that  $\varphi_n\nu = \nu$  (resp.  $\gamma_e\eta = \eta$ ) be satisfied.

We say that a graph  $G(nil, \nu)$  or  $G(\eta, \nu)$  is invariant by the closing if  $\varphi_n\nu = \nu$ . We say that a graph  $G(\eta, nil)$  or  $G(\eta, \nu)$  is invariant by the opening  $\gamma_e$  if  $\gamma_e\eta = \eta$ .

### 3.1.2 Invariant graphs by the closing $\varphi_n$

Consider an arbitrary node weighted graph  $G(nil, \nu)$ . Its node weights are arbitrary and are in general not closed by  $\varphi_n$  unless the following lemma is verified.

**Lemma 15** *The node weights  $\nu$  of a node weighted graph  $G(nil, \nu)$  are invariant by the closing  $\varphi_n$  if and only if the graph has no isolated regional minima.*

**Proof.** For each neighbor  $s$  of  $p$ , we have  $e_{ps} = \nu_p \vee \nu_s \geq \nu_p$ . If  $p$  is not an isolated regional minimum, there exists a neighbor  $q$  of  $p$  verifying  $\nu_p \geq \nu_q$ , we have  $e_{pq} = \nu_p \vee \nu_q = \nu_p$  and  $(\varphi_n \nu)_p = \varepsilon_{pe} \delta_{en} \nu = \nu_p$ . If on the contrary  $p$  is an isolated regional minimum, for each neighbor  $s$  of  $p$ , we have  $e_{ps} = \nu_p \vee \nu_s > \nu_p$  and  $(\varphi_n \nu)_p = \varepsilon_{pe} \delta_{en} \nu > \nu_p$ . ■

If  $G(nil, \nu)$  has isolated regional minima, we add a selfloop edge linking each isolated regional minimum with itself and get a new graph  $\circlearrowleft G_n$  invariant by  $\varphi_n$ . Furthermore, the dilation  $\delta_{en}$  applied to the node weights of  $G_n$  or to the node weights of  $\circlearrowleft G_n$  produce the same edge weights (the relation  $\delta_{en} \varphi_n = \delta_{en}$  shows that only the nodes invariant by  $\varphi_n$  contribute to the dilation  $\delta_{en}$ ).

**Lemma 16** *The graph  $\circlearrowleft G$  is invariant by the closing  $\varphi_n$ . Furthermore it has the same flooding paths, regional minima and catchment zones as the original graph  $G$ .*

**Proof.** Adding loop edges linking isolated regional minima with themselves does not change their nature of being regional minima. It does not either impact the flooding paths. Hence the catchment zones remain the same. ■

On the graph  $\circlearrowleft G(nil, \nu)$  the node function  $\nu$  is invariant by closing:  $\nu = \varphi_n \nu = \varepsilon_{ne} \delta_{en} \nu$ . If we assign to the edges of the graph the weight  $\eta = \delta_{en} \nu$ , then  $\nu = \varepsilon_{ne} \delta_{en} \nu = \varepsilon_{ne} \eta$ . Such a graph is called flooding graph and will be studied below.

Fig.1A presents a node weighted graph  $G(nil, \nu)$  with 3 isolated regional minima. Adding selfloop edges linking each such minimum with itself produces a graph  $\circlearrowleft G(nil, \nu)$  invariant by  $\varphi_n$  in fig.1B. The dilation  $e = \delta_{en} \nu$  produces the edge weights in the flooding graph  $\circlearrowleft G(\delta_{en} \nu, \nu)$  in fig.1C.

### 3.1.3 Invariant graphs by the opening $\gamma_e$

Consider an arbitrary edge weighted graph  $G(\eta, nil)$ . The edge weights  $\eta$  are arbitrary and are in general not open by  $\gamma_e$  unless the following lemma is verified.



**Lemma 17** *An edge weighted graph  $G(\eta, nil)$  with edge weights  $\eta$  is invariant by the opening  $\gamma_e$  if and only if each of its edges is a flooding edge of one of its extremities.*

**Proof.** Suppose the edge  $(p, q)$  is not one of the lowest neighboring edges of  $p$ , i.e.  $(\varepsilon_{ne}\eta)_p < e_{pq}$ . If it is also not the lowest neighboring edge of  $q$ , then  $(\varepsilon_{ne}\eta)_q < e_{pq}$  and  $(\gamma_e\eta)_{pq} = (\delta_{en}\varepsilon_{ne}\eta)_{pq} = (\varepsilon_{ne}\eta)_p \vee (\varepsilon_{ne}\eta)_q < e_{pq}$ . Inversely, if  $(p, q)$  is one of the lowest neighboring edges of one of its extremities, say  $p$ , then  $(\varepsilon_{ne}\eta)_p = e_{pq}$ . And  $(\gamma_e\eta)_{pq} = (\delta_{en}\varepsilon_{ne}\eta)_{pq} = (\varepsilon_{ne}\eta)_p \vee (\varepsilon_{ne}\eta)_q \geq e_{pq}$ . But  $\gamma_e$ , being an opening, is antiextensive:  $\gamma_e\eta \leq \eta$ . So  $(\gamma_e\eta)_{pq} = e_{pq}$ . ■

Hence all edges of  $G(\eta, nil)$  which are not flooding edges for one of their extremities have their weight lowered by  $\gamma_e$ . If we cut these edges, we obtain a new graph  $\downarrow G(\eta, nil)$ , which is invariant by  $\gamma_e$ . Furthermore the erosion  $\varepsilon_{ne}$  applied to the edge weights of  $G(\eta, nil)$  or to the edge weights of  $\downarrow G(\eta, nil)$  produce the same node weights (the relation  $\varepsilon_{ne}\gamma_e = \varepsilon_{ne}$  shows that only the edges invariant by  $\gamma_e$  contribute to the erosion  $\varepsilon_{ne}$ .)

**Lemma 18** *The graph  $\downarrow G$  is invariant by the opening  $\gamma_e$ . Furthermore it has the same flooding paths, regional minima and catchment zones as the original graph  $G$ .*

**Proof.** The edges which are suppressed are the edges which are not the lowest edges of one of their extremities, such edges are never followed by a drop of water falling on a node: they do not belong to any flooding path nor to the regional minima. ■

The operator  $\downarrow$  transforms an arbitrary edge weighted graph  $G(\eta, nil)$  into a graph  $\downarrow G(\eta, nil)$  invariant by the opening  $\gamma_e$ . This means that  $e = \gamma_e e = \delta_{en}\varepsilon_{ne}e$ . If we assign to the nodes of the graph the weight  $n = \varepsilon_{ne}e$ , then  $e = \delta_{en}\varepsilon_{ne}e = \delta_{en}n$ . Again we get a flooding graph.

Fig.2A shows an arbitrary edge weighted graph  $G(\eta, nil)$ , fig.2B the graph  $\downarrow G(\eta, nil)$  in which all edges not invariant by  $\gamma_e$  have been suppressed, and fig.2C the flooding graph  $\downarrow G(\eta, \varepsilon_{ne}\eta)$  with node weights  $\nu = \varepsilon_{ne}\eta$ .

### 3.1.4 Flooding graphs

We already met flooding graphs in the preceding section. We now define and study such graphs.

**Definition 19** *An edge and node weighted graph  $G(\eta, \nu)$  is a flooding graph iff its weights  $(\nu, \eta)$  verify  $\eta = \delta_{en}\nu$  and  $\nu = \varepsilon_{ne}\eta$*

Replacing  $\nu$  by its value we get  $\eta = \delta_{en}\nu = \delta_{en}\varepsilon_{ne}\eta = \gamma_e\eta$ . Replacing  $\eta$  by its value we get  $\nu = \varepsilon_{ne}\eta = \varepsilon_{ne}\delta_{en}\nu = \varphi_n\nu$ . Hence the following corollary.

**Corollary 20** *A flooding graph is invariant by the opening  $\gamma_e$  and by the closing  $\varphi_n$ .*

As established above, as  $\eta$  is invariant by  $\gamma_e$ , each edge is the lowest edge of one of its extremities. The node weights  $\nu$  being invariant by  $\varphi_n$ ,  $G$  has no isolated regional minima.

The preceding section may be summarized by the following corollary.

**Corollary 21** *The graphs  $\circlearrowleft G(\delta_{en}\nu, \nu)$  and  $\downarrow G(\eta, \varepsilon_{ne}\eta)$  are flooding graphs.*

**Flooding edges and flooding paths** Le  $G(\eta, \nu)$  be a flooding graph. A flooding graph has no isolated regional minima and to each node may be associated a flooding edge with the same weight. According to lemma 13 and 14, the flooding edges of the graph  $G(\eta, \nu)$  and of the graph  $G(\eta, nil)$  are identical. Hence the flooding paths, made of successive flooding edges also are identical. As each flooding edge has the same weight as its origin, node weights and edge weights of a flooding path are absolutely redundant.

**Lemma 22** *The regional minima of the graph  $G(\eta, \nu)$  and of the graph  $G(\eta, nil)$  are identical. The first are defined for node weighted graphs, the last for edge weighted graphs.*

**Proof.** The flooding paths of  $G(\eta, \nu)$  and  $G(\eta, nil)$  are identical ; applying the criteria 4 and 8 shows that  $G(\eta, \nu)$  and  $G(\eta, nil)$  have the same regional minima. ■

Having the same flooding paths and the same regional minima, both graphs also have the same catchment zones.

**Theorem 23** *For a flooding graph  $G$ , the flooding paths, regional minima and catchment zones of the graphs  $G(\eta, \nu)$  and  $G(\eta, nil)$  are identical*

**Corollary 24** *An arbitrary node weighted graph  $G(\eta, \nu)$  has the same flooding paths, regional minima and catchment zones as the edge weighted graph  $\circlearrowleft G(\delta_{en}\nu, nil)$ .*

*An arbitrary edge weighted graph  $G(\eta, nil)$  has the same flooding paths, regional minima and catchment zones as the node weighted graph  $\downarrow G(nil, \varepsilon_{ne}\eta)$ .*

## 3.2 Oriented flooding graphs

In fact the flooding paths, regional minima, and catchment zones only depend upon the relative altitudes of neighboring nodes. It is possible to encode these relations in form of an oriented graph and forget the values of nodes and edges.

We associate to the flooding graph  $G$  an oriented graph (or digraph)  $\vec{G}(\eta, \nu)$  by replacing an edge  $e_{pq}$  by an arrow  $\vec{pq}$  if  $e_{pq}$  is a flooding edge of origin  $p$ . If  $e_{pq}$  is a flooding edge of both extremities, we create two arrows  $\vec{pq}$  and  $\vec{qp}$  if  $n_p = n_q$ . The loop edge linking an isolated regional minimum node  $m$  with itself also is replaced by an arrow  $\vec{mm}$ .

The flooding paths of  $G$  simply become the oriented paths of  $\vec{G}$ , which are sufficient for characterizing regional minima or catchment basins. For instance, a regional minimum of  $G$  is characterized in  $\vec{G}$  by the fact that: a) any two nodes are linked by a path along which the successive nodes are linked by two arcs, in both directions ; b) a regional minimum has no incoming arc. Having the same flooding paths and regional minima as the initial graph  $G$ , the oriented graph  $\vec{G}$  also has the same catchment zones.

## 3.3 A visual summary

### 3.3.1 For an initially node weighted graph

Fig.1 presents all avatars of a node weighted graph encountered so far.

A: a node weighted graph  $G(\text{nil}, \nu)$

O: the graph  $G(\delta_{en}\nu, \text{nil})$  ; the graphs  $G(\text{nil}, \nu)$  and  $G(\delta_{en}\nu, \text{nil})$  do not have the same regional minima

B: the graph  $\circlearrowleft G(\text{nil}, \nu)$  obtained by adding to  $G(\text{nil}, \nu)$  loop edges connecting each isolated regional minimum with itself.

C: the flooding graph  $\circlearrowleft G(\delta_{en}\nu, \nu)$

D: the edge weighted graph  $\circlearrowleft G(\delta_{en}\nu, \text{nil})$

E: the flooding digraph  $\vec{\circlearrowleft G}(\delta_{en}\nu, \nu)$

F: the digraph without any weights  $\vec{\circlearrowleft G}(\text{nil}, \text{nil})$

The sign  $\Leftrightarrow$  indicates an inversible transformation preserving regional minima and catchment zones

The sign  $\Rightarrow$  marks an operator preserving regional minima and catchment zones, but is not reversible.

The figures 1 (A,B,C,D,E F) have the same flooding paths, regional minima and catchment basins.

### 3.3.2 For an initially edge weighted graph

Fig.2 presents all avatars of an edge weighted graph encountered so far:

A: an edge weighted graph  $G(\eta, \text{nil})$

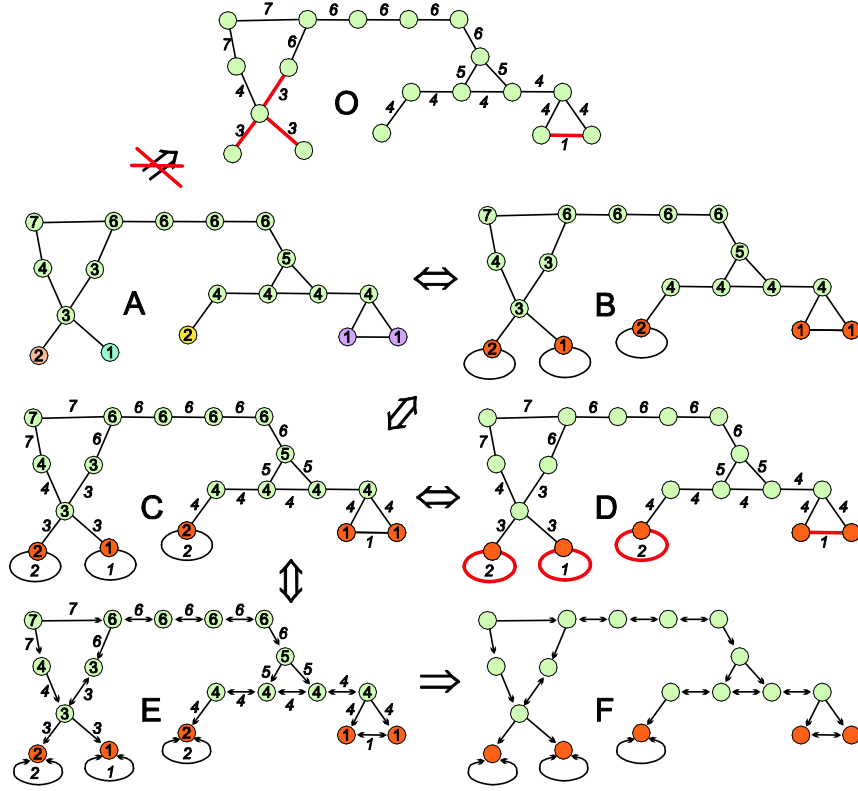


Figure 1: A: a node weighted graph  $G(\text{nil}, \nu)$  ; O:  $G(\delta_{en}\nu, \text{nil})$  ; B:  $\circlearrowleft G(\text{nil}, \nu)$  ; C: the flooding graph  $\circlearrowleft G(\delta_{en}\nu, \nu)$  ; D: the edge weighted graph  $\circlearrowleft G(\delta_{en}\nu, \text{nil})$  ; E: the flooding digraph  $\circlearrowleft \overrightarrow{G}(\delta_{en}\nu, \nu)$  ; F: the digraph without any weights  $\circlearrowleft \overrightarrow{G}(\text{nil}, \text{nil})$

The sign  $\Leftrightarrow$  indicates an inversible transformation preserving regional minima and catchment zones

The sign  $\Rightarrow$  marks an operator preserving regional minima and catchment zones, but is not reversible.

O: the graph  $G(nil, \varepsilon_{ne}\eta)$  has not the same regional minima as the graph  $G(\eta, nil)$

B: the graph  $\downarrow G(\eta, nil)$  obtained by suppressing the non flooding edges of  $G(\eta, nil)$

C: the node and edge weighted flooding graph  $\downarrow G(\eta, \varepsilon_{ne}\eta)$

D: the node weighted graph  $\downarrow G(nil, \varepsilon_{ne}\eta)$

E: the node and edge weighted flooding digraph  $\overrightarrow{\downarrow G}(\eta, \varepsilon_{ne}\eta)$

F: the flooding digraph without weights  $\overrightarrow{\downarrow G}(nil, nil)$

The sign  $\Leftrightarrow$  indicates an inversible transformation preserving regional minima and catchment zones

The sign  $\Rightarrow$  marks an operator preserving regional minima and catchment zones, but is not reversible.

All figures 2(A,B,C,D,E,F) have the same flooding paths, regional minima and catchment basins.

### 3.4 Partial conclusion

It is possible to derive from each node (resp. edge) weighted graph a flooding graph and from the flooding graph an edge (resp. node) weighted graph. A digraph without edge or node weights may also be derived from the flooding graph. All these graphs have the same catchment zones. For this reason, all watershed algorithms developed for node (resp. edge) weighted graphs also apply to edge (resp. node) weighted graphs. It is sufficient to apply them to the derived node (resp. edge) weighted graph.

Furthermore new algorithms, as the one presented below, may be developed applicable to the flooding digraphs.

## 4 Reducing the watershed zones

Each node linked by a flooding path with several distinct regional minima belongs to the corresponding catchment zones. This makes adjacent catchment zones overlap. We aim at reducing these overlapping zones, but without any arbitrary choice between equivalent flooding paths.

Ordering the flooding paths according to their steepness and keeping only the steepest one drastically reduces the overlapping zones of adjacent basins: all edges not belonging to these steepest paths are suppressed. The weights of the nodes or the edges of the graph are not modified. The graph simply has less edges and a drop of water less available trajectories and less freedom to reach distinct regional minima.

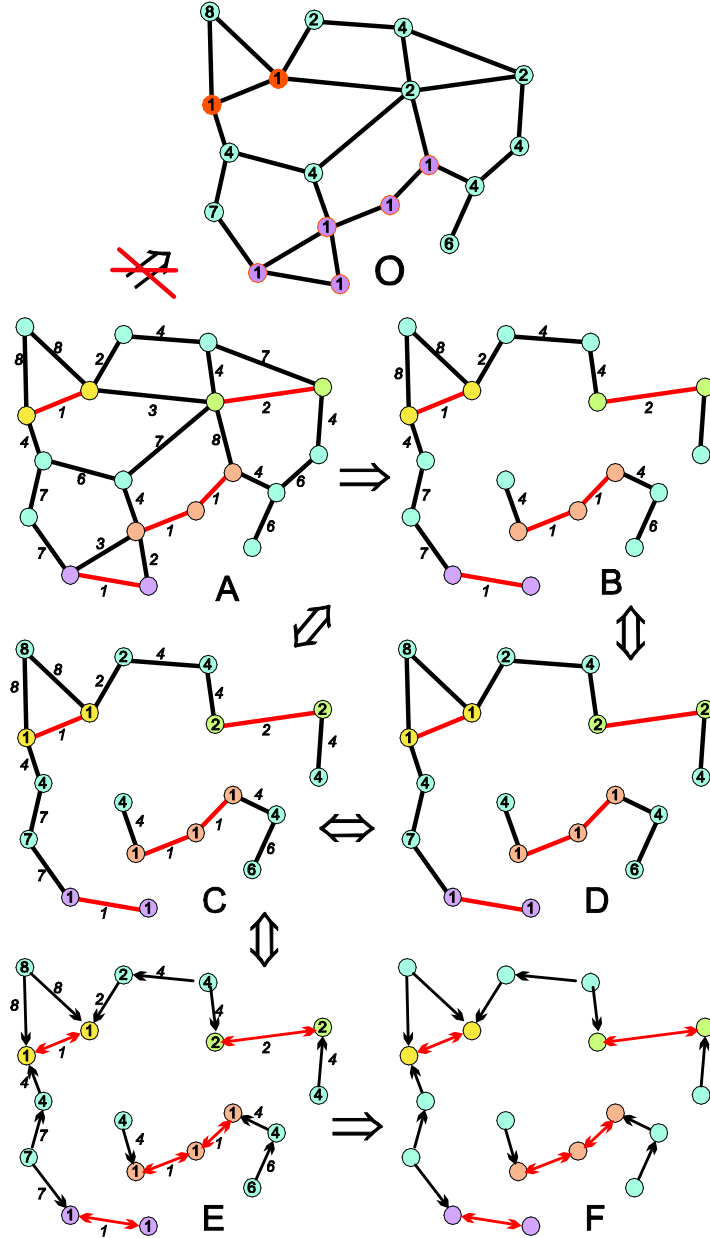


Figure 2: A: an edge weighted graph  $G(\eta, nil)$  ; O:  $G(nil, \varepsilon_{ne}\eta)$  ; B: the graph  $\downarrow G(\eta, nil)$  ; C: the flooding graph  $\downarrow G(\eta, \varepsilon_{ne}\eta)$  ; D:  $\downarrow G(nil, \varepsilon_{ne}\eta)$  ; E: the node and edge weighted flooding digraph  $\downarrow \overrightarrow{G}(\eta, \varepsilon_{ne}\eta)$  ; F: the flooding digraph without weights  $\downarrow \overrightarrow{G}(nil, nil)$

The sign  $\Leftrightarrow$  indicates an inversible transformation preserving regional minima and catchment zones

The sign  $\Rightarrow$  marks an operator preserving regional minima and catchment zones, but is not reversible.

## 4.1 An order relation between flooding paths

Consider a flooding graph  $G(\eta, \nu)$ . A flooding path  $\pi$  containing the series of nodes  $(p_1, p_2, \dots, p_k = m)$  ending at a node  $m$  belonging to a regional minimum may be prolonged into a path of infinite length:

a) if the regional minimum contains another node  $q$ , then the path will indefinitely go back and forth between  $m$  and  $q$  ;

b) if the regional minimum is isolated, it has been completed by a self-loop between  $m$  and itself and the path will indefinitely cycle along this loop.

The **lexicographic preorder relation of length  $k$**  compares the infinite paths  $\pi = (p_1, p_2, \dots, p_k, \dots)$  and  $\chi = (q_1, q_2, \dots, q_k, \dots)$  by considering its  $k$  first nodes and edges:

$$\begin{aligned} * \pi \prec^k \chi & \text{ if } \nu_{p_1} < \nu_{q_1} \text{ or there exists } t < k \text{ such that } \forall l < t : \nu_{p_l} = \nu_{q_l} \\ & \nu_{p_t} < \nu_{q_t} \\ * \pi \preceq^k \chi & \text{ if } \pi \prec^k \chi \text{ or if } \forall l \leq k : \nu_{p_l} = \nu_{q_l}. \end{aligned}$$

## 4.2 The $\widetilde{k - steep}$ flooding paths and $\widetilde{k - steep}$ catchment zones

The preorder relation  $\preceq^k$  is total, as it permits to compare all paths with the same origin  $p$ . The lowest of them are called the  $\widetilde{k - steep}$  paths of origin  $p$ .

If a path  $\pi$  is the  $\widetilde{k - steep}$  of origin  $p$  and if for each node  $q$  of  $\pi$ , the sub-path of origin  $q$  is itself  $\widetilde{k - steep}$  of origin  $q$ , we say that  $\pi$  is uniformly  $\widetilde{k - steep}$  and write  $\widetilde{k - steep}$ . For instance a flooding path is a  $\widetilde{1 - steep}$  path, as each node is followed by one of its lowest adjacent edges (the algorithm for constructing waterfalls in [10] and for constructing watershed cuts in [5] are based on  $\widetilde{1 - steep}$  paths). In a  $\widetilde{2 - steep}$  path, each node is linked by an edge with one of its lowest neighboring nodes ([19, 13] define the catchment basins as Voronoï tessellation of the regional minima for the topographic distance ; the geodesics are then  $\widetilde{2 - steep}$  paths).

The  $\widetilde{k - steep}$  catchment zone associated to a regional minimum  $m_i$  is defined as the set of nodes linked by a  $\widetilde{k - steep}$  path with this minimum. If  $l > k$ , then any  $\widetilde{l - steep}$  path  $\pi$  is also a  $\widetilde{k - steep}$  path. As a consequence, any  $\widetilde{l - steep}$  catchment zone is included in a larger  $\widetilde{k - steep}$  catchment zone. Increasing the index  $l$  reduces the overlapping catchment zones between basins.

An  $\infty - steep$  path of origin  $p$  is at the same time  $\widetilde{\infty - steep}$ , as any of its sub-paths is an  $\infty - steep$  path. Such  $\infty - steep$  catchment

zones rarely overlap, at least if one considers natural images: it only happens if a node is linked with two distinct minima by two flooding paths holding exactly the same weights. In particular, if the regional minima have distinct weights, it cannot happen and the catchment zones are in reality catchment basins and form a partition.

### 4.3 Method by pruning the graph

We now present a local pruning operator which, when applied iteratively, suppresses the highest edge of the paths which are not  $k - \widetilde{\text{steep}}$ ; with each iteration the steepness increases by 1. The pruning is applied on the oriented digraph associated to the initial graph. It makes use of the orientation of the arcs and of the weights of the nodes. The weights of the edges are redundant, as each arc has the same weight as its origin.

We define an erosion and a pruning on the digraph. The erosion  $\vec{\varepsilon}_n$  assigns to each node the weight of its lowest target node (if  $\vec{pq}$  is an arrow, then  $p$  is the origin and  $q$  the target of the arrow). The pruning  $\downarrow^2$  considers all arrows with the same origin and suppresses all those whose target has not a minimal weight.

### 4.4 Keeping only $k - \widetilde{\text{steep}}$ paths

Consider a flooding graph  $\vec{G}$ . We start with the pruning  $\downarrow^2 \vec{G}$ , after which only  $2 - \widetilde{\text{steep}}$  paths survive. We then iteratively apply the operator  $\downarrow^2 \vec{\varepsilon}_n$  to the graph. We call  $\downarrow^k \vec{G}$  the result of applying  $k - 2$  times the operator  $\downarrow^2 \vec{\varepsilon}_n$  to the graph  $\downarrow^2 \vec{G}$ :  $\downarrow^k \vec{G} = (\downarrow^2 \vec{\varepsilon}_n)^{(k-2)} \downarrow^2 \vec{G}$ .

**Theorem 25** *Each arc of  $\downarrow^k \vec{G}$  is the first arc of a  $k - \widetilde{\text{steep}}$  path of the original graph  $\vec{G}$ . We say that  $\downarrow^k \vec{G}$  is a  $k - \widetilde{\text{steep}}$  graph.*

**Proof.** Suppose that after iterating  $(k - 3)$  times the operator  $\downarrow^2 \vec{\varepsilon}_n$  on  $\downarrow^2 \vec{G}$ , we get a  $(k - 1) - \widetilde{\text{steep}}$  graph. Consider now two paths  $\pi = (p_1, p_2, \dots, p_k)$  and  $\pi' = (p_1, p'_2, \dots, p'_k)$  in this graph. The weights of  $p_i$  and  $p'_i$  are  $\nu_i$  and  $\nu'_i$ . The  $k - 1$  first nodes have identical weights but the last ones verify  $\nu_k > \nu'_k$ , indicating that  $\pi'$  is steeper than  $\pi$ . Eroding the graph  $\vec{G}$  once assigns to the node  $p_1$  the weight  $\nu_2 = \nu'_2$  and to the nodes  $p_2$  and  $p'_2$  the weights  $\nu_3 = \nu'_3$ . The subsequent pruning  $\downarrow^2$  preserves the edges  $(p_1, p_2)$  and  $(p_1, p'_2)$ . With successive erosions, the weights of the nodes along the paths  $\pi$  and  $\pi'$  file past the two first nodes. Since the weights of the nodes  $p_2$  and  $p'_2$  are equal after each of these erosions, the operator  $\downarrow^2$  preserves both edges  $(p_1, p_2)$  and  $(p_1, p'_2)$ . The erosion  $k - 2$  assigns to the node  $p_1$  the weight  $\nu_{k-1} = \nu'_{k-1}$  and to the nodes  $p_2$  and  $p'_2$



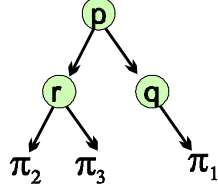


Figure 3: In a  $\widetilde{k - steep}$  graph, each oriented path is a  $\widetilde{k - steep}$  path

respectively the weights  $\nu_k$  and  $\nu'_k$  verifying  $\nu_k > \nu'_k$ . Thus the operator  $|\cdot|^2$  cuts the edge  $(p_1, p_2)$ . The graph is now  $k - steep$ . ■

#### 4.4.1 Properties of $\widetilde{k - steep}$ graphs

In a  $\widetilde{k - steep}$  graph, each arc is the first arc of a  $\widetilde{k - steep}$  path of the original graph  $\vec{G}$ . This means that through each arc  $\vec{pq}$  passes a  $\widetilde{k - steep}$  path with origin  $p$ . The following theorem states a stronger result.

**Theorem 26** *In a  $\widetilde{k - steep}$  graph, each oriented path is a  $\widetilde{k - steep}$  path.*

**Proof.** The theorem is obviously true for  $\widetilde{1 - steep}$  and  $\widetilde{2 - steep}$  graphs. Suppose that it is true for  $(k - 1) - steep$  graphs and let's show it is true for  $\widetilde{k - steep}$  graphs. Consider the fig.3 ; we suppose that it is a  $\widetilde{k - steep}$  graph. Through the arc  $\vec{pq}$  passes a  $\widetilde{k - steep}$  path  $\vec{pq} \triangleright \pi_1$ . Consider another arbitrary oriented path  $\vec{pr} \triangleright \pi_2$  and let us show that it is also a  $\widetilde{k - steep}$  path. The arc  $\vec{pr}$  belongs to a  $\widetilde{k - steep}$  graph and thus is the first arc of  $\widetilde{k - steep}$  path  $\vec{pr} \triangleright \pi_3$ . ( $\pi_2$  and  $\pi_3$  may be identical). But then  $\pi_2$  and  $\pi_3$  are  $\widetilde{2 (k - 1) - steep}$  paths, which by hypothesis are identical. This means that the paths  $\vec{pr} \triangleright \pi_3$  and  $\vec{pr} \triangleright \pi_2$  also are identical. ■

Let us now analyse under which conditions double arcs may subsist in a  $\widetilde{k - steep}$  graph. Consider the fig.4 of a  $\widetilde{k - steep}$  graph. The paths  $\vec{pq} \triangleright \text{first}_{k-1}(\pi_1)$  and  $\pi_2$  are two oriented paths with origin  $p$  of length  $k$  ; hence they are identical. Similarly the paths  $\vec{qp} \triangleright \text{first}_{k-1}(\pi_2)$  and  $\pi_1$  are two oriented paths with origin  $q$  of length  $k$  are identical. We have  $\vec{pq} \triangleright \text{first}_{k-1}(\pi_1) = \pi_2$  and  $\vec{qp} \triangleright \text{first}_{k-1}(\pi_2) = \pi_1$

If the weight of the arcs  $\vec{qp}$  and  $\vec{pq}$  is  $\lambda$  and the weights of the path  $\pi_2$  are  $(\mu_1, \mu_2, \dots, \mu_k)$ , then  $\vec{pq} \triangleright \text{first}_{k-1}(\vec{qp} \triangleright \text{first}_{k-1}(\pi_2)) = (\lambda, \lambda, \mu_1, \dots, \mu_{k-2}) = \pi_2 = (\mu_1, \mu_2, \dots, \mu_k)$ .

From this relation we derive  $\mu_1 = \lambda, \mu_2 = \lambda, \mu_3 = \mu_1 = \lambda, \text{etc.}$  All weights of the paths  $\pi_1$  and  $\pi_2$  are equal. This means that there exists in

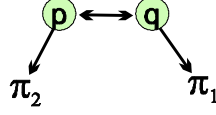


Figure 4: For  $k$  large enough, there are no double arcs anymore in the graph  $\downarrow^k \vec{G}$

the graph  $G$  plateaus with a depth larger than  $k$ . Applying the operator  $\downarrow^k$  did not succeed in letting a smaller value than  $\lambda$  occupy the node  $p$  or the node  $q$ . However, after with the operator  $\downarrow^\infty$ , such a situation cannot happen any more and no plateaus subsist.

## 4.5 Demonstration

Fig.5A (resp. fig.6A) is the same flooding digraph as in fig.1E (resp. fig.2E) in which we only keep the node weights. The successive drawings of fig.5 represent the evolution of the graph during the successive operations, alternatively pruning  $\downarrow^2$  and eroding  $\vec{e}_n$  until stability. After each iteration, the overlapping zones of neighboring catchment zones is reduced. At each iteration, up to convergence the catchment zone get reduced.

## 4.6 Downstream and upstream construction

In a  $\widetilde{k - steep}$  (resp.  $\widetilde{\infty - steep}$ ) graph, each flooding path is a  $\widetilde{k - steep}$  (resp.  $\widetilde{\infty - steep}$ ) path. Following these paths either downstream or upstream permits to segment graphs or parts of graphs. The objects of interest are marked by assigning a label to one or several inside nodes in each object. Distinct objects get distinct labels. Two adaptative operators are used, one for propagating the labels upstream, the other for propagating them downstream.

### Downstream operator

If the node  $p$  holds a label  $\lambda$  : for each node  $q$  such that  $p \rightarrow q$  : assign the label  $\lambda$  to  $q$ .

If several nodes with an arrow towards  $q$  hold distinct labels, one assigns to  $q$  one of them (according to any arbitrary or specific rule).

This operator is applied until stability.

### Upstream operator

If the node  $q$  holds a label  $\lambda$  : for each node  $p$  such that  $p \rightarrow q$  : assign the label  $\lambda$  to  $p$ .

If several nodes with distinct labels are the target of an arrow with origin  $p$ , one of them is assigned to  $p$  (according to any arbitrary or

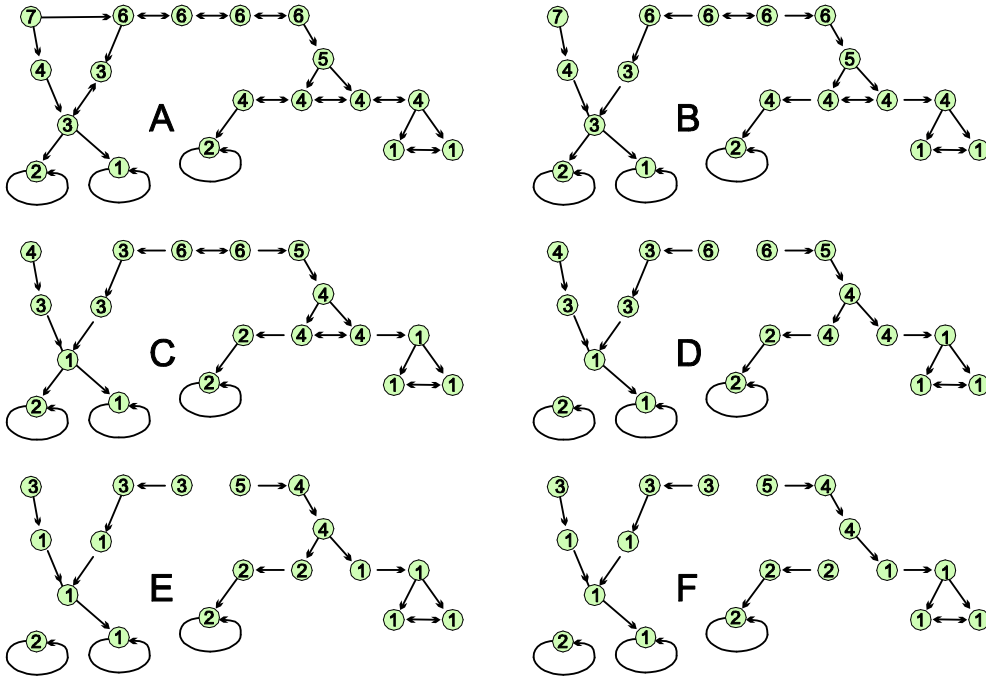


Figure 5: A: an oriented flooding graph  $\overrightarrow{G^{(0)}}$  with its node weights.  
 B:  $\downarrow^2 \overrightarrow{G^{(0)}}$   
 C:  $\overrightarrow{\varepsilon_n} \downarrow^2 \overrightarrow{G^{(0)}}$   
 D:  $G^{(1)} = \downarrow^2 \overrightarrow{\varepsilon_n} \downarrow^2 \overrightarrow{G^{(0)}} = \zeta \downarrow^2 \overrightarrow{G^{(0)}}$   
 E:  $\overrightarrow{\varepsilon_n} G^{(1)}$   
 F:  $G^{(2)} = \downarrow^2 \overrightarrow{\varepsilon_n} G^{(1)} = \zeta G^{(1)}$ . The resulting graph is cut in 4 connected components, each containing one regional minimum.

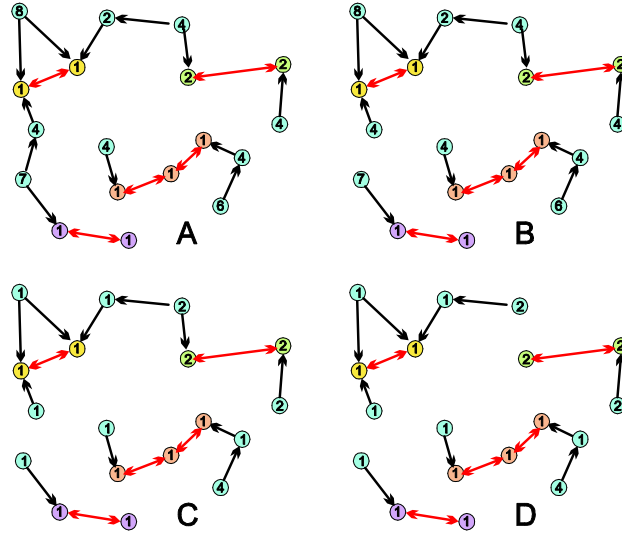


Figure 6: A: an oriented flooding graph  $\overrightarrow{G^{(0)}}$  with its node weights.  
 B:  $\downarrow^2 \overrightarrow{G^{(0)}}$   
 C:  $\overrightarrow{\varepsilon_n} \downarrow^2 \overrightarrow{G^{(0)}}$   
 D:  $\overrightarrow{G^{(1)}} = \downarrow^2 \overrightarrow{\varepsilon_n} \downarrow^2 \overrightarrow{G^{(0)}} = \downarrow^3 \overrightarrow{G^{(0)}}$ . The resulting graph is cut in 4 connected components, each containing one regional minimum.

specific rule).

This operator is applied until stability.

**Remark**

It is the first time that we meet an arbitrary choice in this paper. This choice occurs if there is a conflict between several downstream or upstream regions with distinct labels.

#### 4.6.1 Extraction of one or marked catchment zones

The downstream and upstream operators permit to selectively extract one or several catchment zones in an image or in a graph. A flooding graph is first produced, transformed in its flooding digraph and then pruned in order to extract a  $k - \widetilde{steep}$  or  $\infty - \widetilde{steep}$  digraph.

One or several nodes are then assigned labels. These labels are first propagated downstream ; after convergence, the resulting labeled nodes contain the regional minima of the labeled catchment zones ; their upstream propagation extracts the marked catchment zones.

Fig.7 presents a digital elevation model of a landscape containing multiple rivers belonging to distinct geographical catchment basins. A node has been marked in the left image. The downstream of this node is a river and a regional minimum, illustrated in the central image. The upstream of this river constitutes the catchment zone of the marked node.

As the value  $k$  increases, the catchment zones get smaller and become closer and closer to the ideal catchment zones associated to  $\infty - \widetilde{steep}$ . Already small values of  $k$  give acceptable approximations of catchment zones. In fig.8 we compare the catchment zones extracted from the same flooding digraph after only a few steps of pruning. The first one has been obtained for  $\downarrow^3 \vec{G}$ , the second for  $\downarrow^4 \vec{G}$ , the third for  $\downarrow^6 \vec{G}$ . This last one is almost identical with the catchment zones associated to  $\downarrow^\infty \vec{G}$ . As the operator  $\downarrow^k$  is extremely simple, it may be easily implemented in any architecture. Therefore, it is not costly to apply it to the flooding digraph for small values of  $k$ . A parallel algorithm as in [12] will give better results as the original paper which only considers  $2 - \widetilde{steep}$  graphs.

The  $\infty - \widetilde{steep}$  paths are extremely narrow. In fig. 9 we have marked a few minor rivers. The downstream operator nicely and selectively extracts the complete river having the marked points as origins.

## 5 Conclusion

Considering overlapping catchment zones in place of a partition of catchment basins has opened new perspectives. Representing the topographic

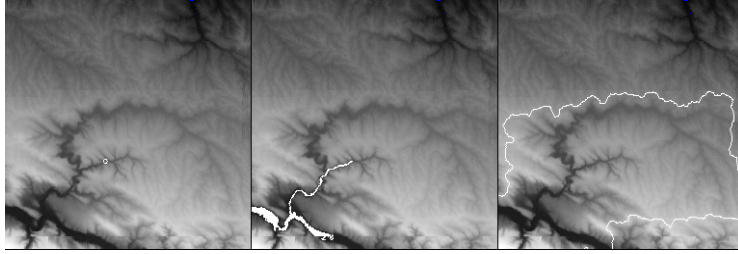


Figure 7: Left: a node  $p$  is marked ; Centre: Downstream of the node ; Right: Upstream of the downstream=catchment zone containing  $p$

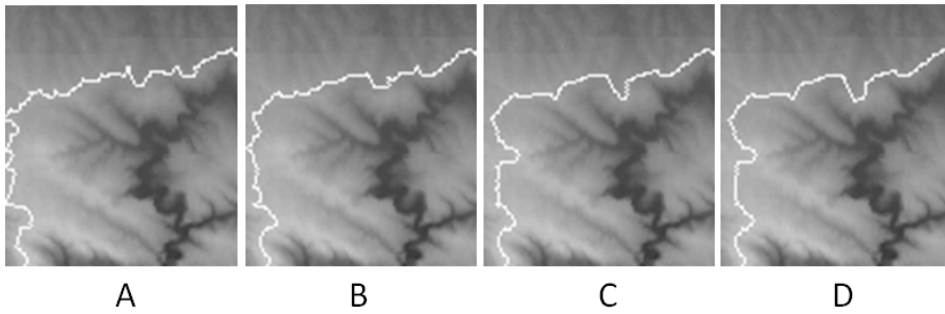


Figure 8: Extraction of a marked catchment zone. Details showing the evolution of the contour after respectively 1, 2, 4 and an infinite number of pruning steps.

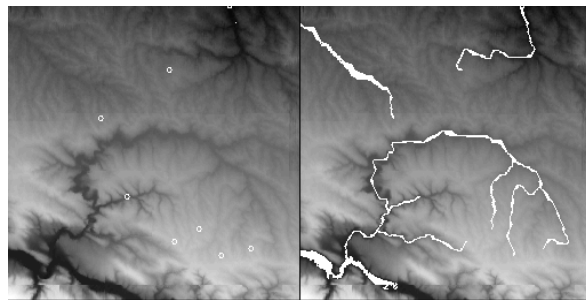


Figure 9: Left: a number of pixels are marked  
Right: Downstream of the markers = rivers

surface as weighted graphs permits to address both the construction of the watershed on images as the construction of the waterfall or watershed on region adjacency graphs.

In a flooding graph, the flooding paths, regional minima and catchment basins remain the same whether one considers the node weights only or the edge weights only. We have shown how to transform any node or edge weighted graph into a flooding graph. Like that, all algorithms developed for node weighted graphs or for edge weighted graphs may be applied on the flooding graph.

By reducing the number of available trajectories a drop of water may follow, the catchment zones also overlap less. An iterative pruning algorithm progressively suppresses flooding paths according to their steepness. According to the degree of pruning, approximative catchment basins or exact catchment basins may then be obtained, with minimal arbitrary choices.

As the remaining flooding paths are extremely selective, they lead to new applications: following the downstream or the upstream of a node, studying the local orientation of the steepest paths, constructing adaptive neighborhoods for filtering images.

**Acknowledgement 27** *I am grateful for the fruitful discussion with Jean Serra on an early version of this manuscript and for his suggestions for improvements.*

## References

- [1] Romaric Audigier and Roberto Lotufo. Uniquely-determined thinning of the tie-zone watershed based on label frequency. *Journal of Mathematical Imaging and Vision*, 27(2):157–173, 2007.
- [2] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. *ISMM94 : Mathematical Morphology and its applications to Signal Processing*, pages 69–76, 1994.
- [3] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *Proc. Int. Workshop Image Processing, Real-Time Edge and Motion Detection/Estimation*, 1979.
- [4] Moga A. Bieniek, A. An efficient watershed algorithm based on connected components. *Pattern Recognition*, 33(6):907–916, 2000. cited By (since 1996) 78.
- [5] Jean Cousty, Gilles Bertrand, Laurent Najman, and Michel Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1362–1374, 2009.

- [6] J. Cousty, L. Najman, J. Serra, Some morphological Operators on Graph Spaces in *Mathematical Morphology and its Applications to Signal and Image Processing (Proceedings of ISMM 2009)*, M.H.F. Wilkinson and B.T.M. Roerdink, Eds, Springer 2009, pp.149-160
- [7] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed Cuts: Thinnings, Shortest Paths Forests, and Topological Watersheds, in *IEEE PAMI*, vol.32, n° 5 pp. 925-939, May 2010
- [8] A.X. Falcao, J. Stolfi, and R. de Alencar Lotufo. The image foresting transform: theory, algorithms, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1):19 – 29, jan 2004.
- [9] F. Maisonneuve. Sur le partage des eaux. *Technical Report-Centre of Mathematical Morphology-Mines-ParisTech*, 1982.
- [10] B. Marcotegui and S. Beucher. Fast implementation of waterfalls based on graphs. *ISMM05 : Mathematical Morphology and its applications to Signal Processing*, pages 177–186, 2005.
- [11] F. Meyer and S. Beucher. Morphological segmentation. *JVCIP*, 1(1):21–46, Sept. 1990.
- [12] F. Meyer. Un algorithme optimal de ligne de partage des eaux. In *Proceedings 8<sup>ème</sup> Congrès AFCET, Lyon-Villeurbanne*, pages 847–857, 1991.
- [13] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, pages 113–125, 1994.
- [14] F. Meyer and J. Angulo. Micro-viscous morphological operators. *ISMM07 : Mathematical Morphology and its applications to Signal and Image Processing*, pages 165–176, 2007.
- [15] Fernand Meyer and Jean Stawiaski. Morphology on graphs and minimum spanning trees. In *Mathematical Morphology and Its Application to Signal and Image Processing*, pages 161–170. Springer, 2009.
- [16] F. Meyer. The steepest watershed: from graphs to images. *arXiv preprint arXiv:1204.2134*, 2012.
- [17] F. Meyer. Watersheds, waterfalls, on edge or node weighted graphs. *arXiv preprint arXiv:1204.2837*, 2012.
- [18] F. Meyer. The watershed concept and its use in segmentation : a brief history. (*arXiv:1202.0216v1*), 2012.
- [19] L. Najman and M. Schmitt. Watershed of a continuous function. *Signal Processing*, 38(1):99 – 112, 1994. *Mathematical Morphology and its Applications to Signal Processing*.
- [20] L. Najman and H. Talbot (ed) *Mathematical morphology Wiley editor*, 2012
- [21] J. B. T. M. Roerdink and A. Meijster. The watershed transform:



- Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2001.
- [22] J. Serra, editor. *Image Analysis and Mathematical Morphology. II: Theoretical Advances*. Academic Press, London, 1988.
- [23] Soille Pierre Vincent, Luc. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.